Network Services

Jun 5, 1989

Introduction

The Acnet network header will be used by D0 for task-to-task communication across the network. The current Acnet services use the concept of a user-specified reply buffer to receive replies to a request message. An alternate scheme is described here.

It is desired to make use of pSOS message queues (also called exchanges) to pass network messages through the system. In this case, the reply buffer is not allocated directly by the user but is allocated by the network software in the form of a circular buffer. A received message is passed to the user via a pointer. This saves the overhead of copying the received message, and it allows for multiple reply messages to be queued up awaiting processing by the requester. Reply messages cannot be overwritten by new replies that arrive before the requester sees them. Furthermore, an AST does not have to be utilized to get around this latter problem.

pSOS Message Queues

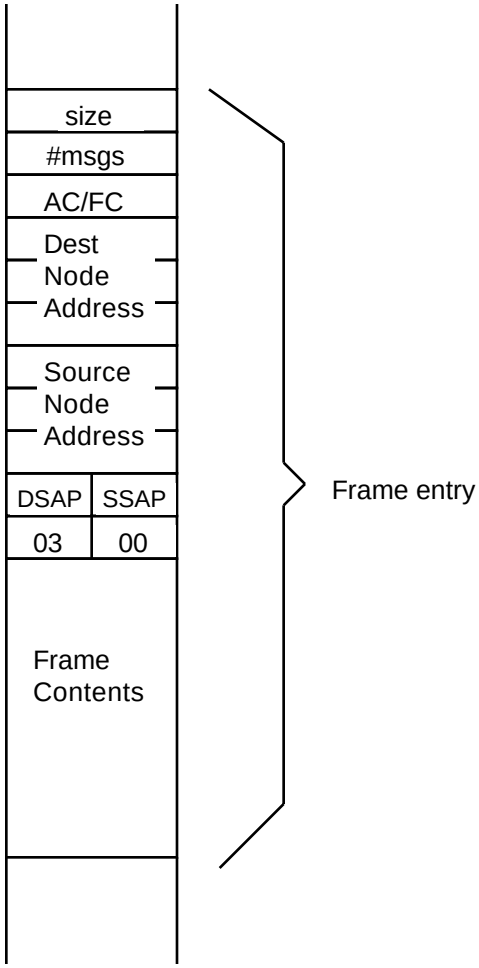
Under pSOS, a message queue entry is composed of 4 longwords (16 bytes). The queue is identified by a 4-byte name and also by a 4-byte id. A long message must therefore include a pointer to the *real* message in the message queue entry. Through pSOS service calls, a task can send a message to a queue. It can also wait on the queue for a message to arrive. It does this by reading from the queue; if the queue is not empty, the message at the head of the queue is returned. If the queue is empty, the task may elect to return immediately with status, or it may elect to be suspended until a message arrives, whereupon it will be made ready.

When a queue is created, the queue id is returned for use with most service calls which refer to the queue. If another task wishes to use the queue, it can make an attach call with the queue's name to get the queue id. When the queue is no longer needed, it can be deleted.

User Network Access

When a task wishes to use the network, it should call NetConnect to connect to the network, giving the taskname and the queue id of a message queue it has created. When a message is received from the network with the proper SAP for Acnet-header messages, it is passed through the message queue for processing.

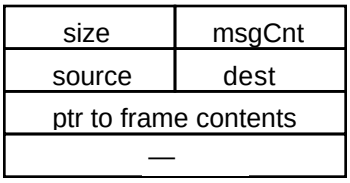
they have been received into system memory. A circular buffer is used to receive network frames. As there is no way to know how long a frame is before it is received, the end of the circular buffer may not get much use. Each frame's entry has the following format:



Everything but the first two words comes by DMA from the token ring chip set. The size word is the total size of the frame entry = frameSize+4. The #msgs word contains a count of the number of messages in the frame. It is decremented by each task which processes the messages. When it is reduced to zero, all messages have been processed in that frame entry, and the space can be made available for additional frame entries. The network receive interrupt must manage the space in the circular buffer, as it pre-pares the buffer pointer and count values used in the receive parameter list.

The receive interrupt routine checks for a valid received frame. It examines the AC/FC fields, and it checks for the DSAP and $03 control byte values. Based upon the DSAP value, it sends a frame message to the message queue to be passed to the task which

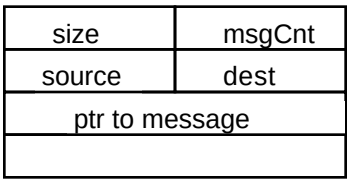
message is as follows:



The size value is the size of the frame contents *only*; the length of the frame header has been removed from the received frame size. The source, dest, and msgCnt values are offsets (from the frame contents pointer) to the relevant fields in the frame entry.

The task which processes these messages (for a given frame format as determined by the DSAP value) must distribute the messages it finds to the appropriate task(s) to handle them. Each frame in general contains multiple messages, and each message header may include a destination task name or source task id that is different from other messages in the same frame. The network hardware delivers frames between nodes; it knows nothing about tasks within nodes.

The frame processing task, such as the Acnet Task, scans the frame contents for messages. Based upon the destination task name (used for request messages and USM's) or the source task id (used for replies), it delivers each message to the message queue appropriate to the designated task. The size of each message is included in the Acnet header in the 9th word. The format of the delivered message queue entry delivered to the designated task is:



The size word is the size of the *message*, including any Acnet header, format block and the message itself. The source, dest, and msgCnt refer to the offsets (from the message pointer) to the relevant fields of the frame entry in the circular buffer.

A table called NADDR is used to keep the 6-byte network node addresses of each node. Internally, each node on the network is denoted by a single byte value. The NADDR table is indexed by this internal node#. Each entry in this table consists of 8 bytes as follows:

6-byte network node address	count

When the task handling a particular SAP format receives a request message or a USM (but not a reply message), it checks the entry in this table indexed by source node#. If the source address (from the frame header) matches, the count is incremented. If it does not match, the new source address is entered, and the count is initialized to 1. When a reply message is to be delivered to the source node, the entry in this table is used to prepare the destination network node address for use in the frame header by the network transmit logic. The last few entries of this table contain the group addresses which can be used. The last entry, indexed by node address $FF, denotes the broadcast group address. Other entries, from $FE on down, may be used to denote various functional group addresses.

When one of the message handling tasks, which receives a message pointer queue entry, has processed the message, it should decrement the msgCnt word associated with the frame. This allows the network interrupt routine, when the count reaches zero, to advance the OUT queue pointer to reflect the new space available for more received network frames.